CLAIMS

1. Code generator (60) for generating an orthogonal code having a spreading factor SF and an index k, wherein the spreading factor SF is selectable from values in a range $1 < SF \leq SF_{max}$ with SF_{max} denoting a maximum spreading factor, said code generator (60) including:

   a) an index conversion unit (61) for converting said index k into a modified index j associated with a corresponding code having the maximum spreading factor,
   b) a logic unit (62) for performing logic operations on bits of said modified index j and bits of a counter value i, thereby generating a code bit of said orthogonal code.

2. Code generator according to claim 1, wherein said corresponding code is one of: an OVSF code, a Hadamard code, a Walsh code.

3. Code generator according to claim 1 or 2, wherein said index conversion unit (61) includes multiplication means (71,72) for multiplying said index k with a value of SF_{max}/SF.

4. Code generator according to claim 3, wherein said multiplication means (71,72) includes:
   - a mapping unit (72) for mapping said spreading factor SF to a number s equal to $\log_2\{SF_{max}/SF\}$,

- a shift register (71) adapted to receive and store said index k in binary representation, further adapted to receive said number s and to shift the stored index k by s bit positions in the direction of more significant bit positions.

5. Code generator according to one of the preceding claims, wherein said index conversion unit (61) includes a permutation unit (73) for permuting the bits of said index k.

6. Code generator according to claim 3 or 4, wherein said index conversion unit (61) includes
- a permutation unit (73) for permuting the bits of said index k,
- a selection means (74) for selecting, in dependence of a mode signal indicating a desired type of said orthogonal code, the output of said permutation unit (73) or the output of said shift register (71), thereby generating said modified index j.

7. Code generator according to one of the preceding claims, wherein said logic unit (62) includes:
- adding means (81-1,…,81-9) for performing binary AND operations, wherein each adding means is adapted to receive a bit of said modified index j and a bit of said counter value i, and is further adapted to output a binary output value representing a binary AND combination of said two bits,
- combining means (82) for combining said binary

output values into said code bit.

8. Code generator according to claim 7, wherein said combining means (82) includes means for performing binary XOR operations (82-1,…,82-8).

9. Code generator according to one of the preceeding claims, further including a counter (63) for generating said counter value i.

10. Parallel code generator (90) for concurrently generating p>1 orthogonal codes having respective spreading factors $SF_1,\dots,SF_p$ and indices $k_1,\dots,k_p$, wherein the spreading factors are selectable from values in a range $1 < SF_1,\dots,SF_p \leq SF_{max}$ with SF_{max} denoting a maximum spreading factor, said parallel code generator (90) including:

a) p code generators (90-1,90-2,…,90-p) according to one of the claims 1 to 8, each for generating one of said p orthogonal codes having a particular one of said spreading factors and a particular one of said indices,

b) a counter (93) for generating said counter value i to be used by said p code generators.

11. Parallel code generator for concurrently generating p>1 orthogonal codes having respective spreading factors $SF_1,\dots,SF_p$ and indices $k_1,\dots,k_p$, wherein the spreading factors are selectable from values in a range $1 < SF_1,\dots,SF_p \leq SF_{max}$ with SF_{max} denoting a

maximum spreading factor, said parallel code generator including
p code generators according to claim 9, each for generating one of said p orthogonal codes having a particular one of said spreading factors and a particular one of said indices.

12. Code generation method for generating an orthogonal code having a spreading factor SF and an index k, wherein the spreading factor SF is selectable from values in a range $1 < SF \leq SF_{max}$ with SF_{max} denoting a maximum spreading factor, said code generation method including the steps of:

a) converting (101) said index k into a modified index j associated with a corresponding code having the maximum spreading factor,
b) initialising (102) a counter value i,
c) performing logic operations (103) on bits of said modified index j and bits of said counter value i, thereby generating a code bit of said orthogonal code,
d) incrementing (104) said counter value i by one,
e) repeating said step of performing logic operations (103) and said step of incrementing (104) until a desired number of code bits has been generated.

13. Code generation method according to claim 12, wherein said corresponding code is one of: an OVSF code, a Hadamard code, a Walsh code.

14. Code generation method according to claim 12 or 13, wherein said step of converting (101) includes a step of multiplying (111,112,113) said index k with a value of SF_{max}/SF.

15. Code generation method according to claim 14, wherein said step of multiplying (111,112,113) includes the steps of:
- mapping (111) said spreading factor SF to a number s equal to $\log_2\{SF_{max}/SF\}$,
- storing (112) said index k in binary representation in a shift register,
- shifting (113) the stored index k by s bit positions in the direction of more significant bit positions.

16. Code generation method according to one of the claims 12 to 15, wherein said step of converting (101) includes a step of permuting (114) the bits of said index k.

17. Code generation method according to claim 14 or 15, wherein said step of converting (101) includes the steps of:
- permuting (114) the bits of said index k,
- selecting (115), in dependence of a mode signal indicating a desired type of said orthogonal code, the permuted index or the shifted index, thereby generating said modified index j.

18. Code generation method according to one of the claims 12 to 17, wherein said step of performing logic operations (103) includes the steps of:
- performing binary AND operations (121), wherein each operation is adapted to combine a bit of said modified index j and a bit of said counter value i, and to output a binary output value representing a binary AND combination of said two bits,
- combining (122) said binary output values into said code bit.

19. Code generation method according to claim 18, wherein said step of combining (122) includes a step of performing binary XOR operations.

20. A computer program product directly loadable into an internal memory of a communication unit, comprising software code portions for performing the steps of one of the claims 12 to 19, when the product is run on a processor of the communication unit.